

## PLP - Recuperatorio del Segundo Parcial - 2<sup>do</sup> cuatrimestre de 2022

Este examen se aprueba obteniendo al menos dos ejercicios bien menos (B-) y uno regular (R). Las notas para cada ejercicio son: -, I, R, B-, B. Entregar cada ejercicio en hojas separadas. Poner nombre, apellido y número de orden en todas las hojas, y numerarlas. Se puede utilizar todo lo definido en las prácticas y todo lo que se dio en clase, colocando referencias claras. El orden de los ejercicios es arbitrario. Recomendamos leer el parcial completo antes de empezar a resolverlo.

### Ejercicio 1 - Programación Lógica

- a) Definir el predicado `sublistaMasLargaDePrimos(+L,?P)` que es verdadero cuando P es una sublista de L que contiene la mayor cantidad de números primos contiguos. Puede haber más de una solución. Por ejemplo:

```
?- sublistaMasLargaDePrimos([2,3,4,5,6,13,7],S).  
S = [2,3];  
S = [13,7];  
false.
```

- b) Dado el predicado `simbolo(?S)` que instancia en S ciertos elementos de un alfabeto, se pide definir el predicado `clausura(-L)` que es verdadero cuando L es la lista que representa la clausura de Kleene<sup>1</sup> del lenguaje conformado por los símbolos S. Por ejemplo, si a y b son los únicos elementos que hacen verdadero a `simbolo`, entonces:

```
?- clausura(L).  
L = [];  
L = [a];  
L = [b];  
L = [a,a];  
L = [a,b];  
L = [b,a];  
L = [b,b];  
L = [a,a,a];  
L = [a,a,b];  
...
```

- c) El predicado del inciso anterior, ¿es reversible en L? Justificar.

### Ejercicio 2 - Resolución

Considerar las siguientes definiciones en prolog:

```
insertar(X,L,LX) :- append(L1L2D), append(L1,[X|L2],LX). ->
```

```
+ append([],L,L).
```

```
+ append([X|XS],YS,[X|L]) :- append(XS,YS,L).
```

y la consulta:

```
?- insertar(3,[1,2],[1,3,R]).
```

- a) Convertir la base de conocimientos y la consulta a forma clausal.

Sugerencia: pueden pensar la lista [1,2] como [1|[2|[]]]

- b) Utilizar el método de resolución para obtener el resultado de la consulta.

- c) La resolución utilizada en el inciso anterior, ¿fue SLD? Justificar.

<sup>1</sup>La Clausura de Kleene es una operación de los lenguajes que representa el conjunto de cadenas que se pueden formar con los símbolos del lenguaje. Las cadenas pueden ser vacías e incluir repeticiones.

### Ejercicio 3 - Objetos

a) Sea el siguientes código escrito en JavaScript:

```
let a = { v: 1, f: function(x){ return this.v + x; } };
let b = Object.create(a);
let c = { v: 2 };
Object.assign(b, c);
console.log(b.f(3));
```

Indicar qué mensajes se envían durante la ejecución de la última línea, cuál es el objeto receptor de cada mensaje, y qué valor se imprime en la consola. (La suma no se considera un mensaje, ya que en JavaScript los números no son objetos).

b) Sean  $a$ ,  $b$  y  $c$  los siguientes objetos del cálculo Sigma:

$$a \stackrel{\text{def}}{=} [ v = \text{uno}, f = \lambda(z) \text{ suma } (z.v) (x) ]$$
$$b \stackrel{\text{def}}{=} [ v = \text{dos}, f = a.f ]$$
$$c \stackrel{\text{def}}{=} [ v = \text{dos} ]$$

Decidir si estos objetos se comportan de la misma manera que los correspondientes  $a$ ,  $b$  y  $c$  del punto anterior, luego de haber ejecutado la línea del `Object.assign()`. Justificar.

Mostrar cómo reduce la expresión  $b.f(\text{cero})$ . Se puede usar la regla APP vista en clase. Asumir que  $\text{suma}(\text{uno})(\text{cero}) \rightarrow \text{uno}$  y  $\text{suma}(\text{dos})(\text{cero}) \rightarrow \text{dos}$ . Consideramos que  $\text{cero}$ ,  $\text{uno}$  y  $\text{dos}$  son valores.

$$\frac{f \rightarrow g \quad g \equiv \lambda(x)t \quad t\{x \leftarrow a\} \rightarrow v}{f(a) \approx v} [\text{APP}]$$



mp: 7

Ej	1	2	3
	B-	R-	B-

Con  
Dona

Bocaccio Sebastián

Aprobado Condicional

hoja 1

→ esto en la hoja

EJERCICIO 1:

a) ~~EJERCICIO 5V, 10 puntos ya dados~~

% subestomasLaresPrimos(+L, ?P)

subestomasLaresPrimos(L, P) :-  $\frac{\text{sublistas}(L, P), \text{todasPrimas}(P)}{\text{moE}(\text{otras subestomasLares}(L, P))}$

% todasPrimas(+P)

todasPrimas(0).  $\text{moE}(\text{moEPrimas}(X))$

todasPrimas([X|XS]) :-  $\text{moE}(\text{moEPrimas}(X))$ , todasPrimas(XS)

código  
repetido

% otras subestomasLares(+L, +P).

otras subestomasLares(L, P) :-  $\frac{\text{sublistas}(L, P)}{\text{todasPrimas}(P)}$ ,  $\text{between}(P_1, N_1), \text{between}(P_2, N_2)$   
 $N_1 > N_2$ .

~~otras subestomasLares(+L, +P).~~

% ~~moEPrimas(X)~~

~~Primo(X) :-  $\text{between}(2, N, X)$ ,  $\text{between}(X, N, X)$ ,  $1 = 1 = \text{gcd}(X, X)$ .~~

~~moEPrimas(X) :-  $\text{between}(2, N, X)$ ,  $\text{between}(X, N, X)$ ,  $1 = 1 = \text{gcd}(X, X)$ .~~

!esto es verdadero para  $X=1$ .



7  
b)  $\neg$  clausura ( $\neg L$ ).

clausura( $L$ ) :-  $\text{length}(L, N)$ , todos los elementos son símbolos( $L$ ).

$\neg$  todos los elementos son símbolos( $L$ )

todos los elementos son símbolos( $L$ )

todos los elementos son símbolos( $[X|XS]$ ) :- símbolo( $X$ ), todos los elementos son símbolos( $XS$ )

c) Para que el predicador sea reversible debe pasarle una lista, y que si todos los elementos son símbolos devuelve true y sino false.

Efectivamente lo es ya que si  $L$  esta instanciada  $\text{length}(L, N)$  sigue funcionando correctamente. Luego, todos los elementos son símbolos es reversible si símbolo lo es, por lo tanto, concluimos que es reversible.



No. 2

EJERCICIO 2.

o)

$$\text{insert}(x, L, Lx) :- \text{append}(L, Lx, L), \text{append}(L, Lx, Lx)$$
 $\equiv$ 

$$\text{insert}(\text{append}(L_1, L_2, L) \vee \text{append}(L_1, [x|L_2], L) = \text{insert}(x, L, Lx)$$

$$(\neg \text{append}(L_1, L_2, L) \vee \neg \text{append}(L_1, [x|L_2], L) \vee \text{insert}(x, L, Lx)$$
~~At. 1~~  $L_2$ 

$$1) \{ \neg \text{append}(L_1, L_2, Lx), \neg \text{append}(L_1, [x|L_2], Lx),$$

$$\text{insert}(x, L_1, Lx) \}$$

$$2) \{ \text{append}(L_2, L_1, Lx) \}$$
~~At. 2~~

$$\text{append}([x|L_1], L_2, [x|L_2]) :- \text{append}(L_1, L_2, L_2)$$
 $\equiv$ 

$$\text{append}(L_1, L_2, L) \Rightarrow \text{append}([x|L_1], L_2, [x|L_2])$$

$$\neg \text{append}(L_1, L_2, L) \vee \text{append}([x|L_1], L_2, [x|L_2])$$

$$3) \{ \neg \text{append}(L_1, L_2, L), \text{append}([x|L_1], L_2, [x|L_2]) \}$$
consulta: Dado que  $L_1, L_2 = [1, 2, 3]$ 

$$\{ \text{insert}(3, [1, 2, 3]) \}$$



b) ~~como lo dice~~

No nos dividamos de nuevo la consulta porque sino no lo voy a poder hacer.

4)  $\{ \text{insertar}(3, [1 | [2 | []]], [1, 3, 2]) \}$   $[1 | [2 | [3 | [R | []]]]]$

Ahora bien, entonces de armar un plan  $\rightarrow$  no sé cómo llamar a  $L_1 = [1]$ ,  $L_2 = [2]$

me escribir las reglas de nuevo

1)  ~~$\{ \text{append}(L_a, L_b, L_c), \text{append}(L_a, [x] | L_b, L_c), \text{insert}(x, L_a, L_b) \}$~~

2)  $\{ \text{append}([], L_b, L_b) \}$

3)  $\{ \text{append}(x_{sc}, y_{sc}, L_c), \text{append}([x | x_{sc}], y_{sc}, [x_c | L_c]) \}$

4)  $\{ \text{insert}(3, [1 | [2 | []]], [1, 3, 2]) \}$



hoid 3  
BOOCCUR SEBASTIAN

LRPS K2002  
R=2

hoid 3  
LACY  
K2002

Insertor :-  $\text{append}(L_1, L_2, (1, 2))$ ,  $\text{append}(L_2, [3|L_2], [1, 3, 2])$

1 → fail

$([] , [1, 2])$ ,  $\text{append}([], [1, 2], [1, 2])$ ,  $\text{append}([], [3|L_2], [1, 3, 2])$  → 2  
 $\text{append}([], [1, 2], [1, 2])$ ,  $\text{append}([], [3, 2], [1, 3, 2])$

2

→  $\text{append}([], [3, 2], [3, 2])$  COM 1 zero s. node has elem on R.

Insertor  
- 1st  $\text{append}([], [1, 2], [1, 2])$ ,  $\text{append}([], [3, 2], [1, 3, 2])$

Do 1 y4 (another code for)

$x_0 \in 3$ ,  $L_0 \in [1|L_1|L_2]$ ,  $L_{x_0} \in [1, 3, 2]$

5)  $\{ \neg \text{append}(L_1, L_2, [1|L_2|L_3]) \}$ ,  $\neg \text{append}(L_1, [3|L_2], [1, 3, 2])$

Do 3 y5 en  $\{ x_0 \in 1, L_0 \in [2|L_1] \}$ ,  $L_{x_0} \in$



Ejercicio 3 :

Simplemente, en este momento no tengo si quiero un objeto  
 Max/Min no sabe responder a un msg, pero si le sabe responder  
 a su protocolo a quien corresponde el "LHS" si el objeto original  
 o a su protocolo, Recuerda que Max/min le compite por siempre tener  
 que volver a ~~ser~~ the queue a hacer porque me olvidé.  
 Nos a examinar ambos casos.

- El primer mensaje que se envía es LOS con receptor consele ✓
  - ~~El objeto Max/Min~~ <sup>interviene</sup> ~~conviene~~ ~~conviene~~ ~~los~~ ~~que~~ ~~a~~
  - El segundo mensaje que se envía ~~Max/Min~~ <sup>Busca</sup> ES f y el receptor es d objeto b
  - Como b me sabe responder f, ~~Max/Min~~ <sup>Busca</sup> el mensaje a su protocolo "a"
  - El siguiente mensaje es N y en este caso es Z. ~~que es un mensaje~~ X
- El otro que se incluye es el resultado de enviar Z + 3, o sea 5. ✓



b) Não no SE conforto do b mesmo mesmo.  
Em este caso, si b infante aplica f com 3 caso período, o  
resultado na 3 ser 4. este as así 32 que d valor n que se  
utiliza as d do 2 y no el De b.



no: 7  
BOGOTÁ SEBASTIÁN

hoja 5

funcion <sup>def</sup> = ~~obj~~  $\lambda(x)$  sumo(2.v) (x).

$\frac{\text{obj}}{b \rightarrow b}$	$\frac{\frac{\text{obj}}{a \rightarrow a} \quad \frac{\text{funcion} \rightarrow \text{funcion}}{\text{sel}}}{a.f \rightarrow \text{funcion}}$	$\frac{\frac{\text{obj}}{2 \rightarrow 2} \quad \frac{\text{funcion} \rightarrow \text{funcion}}{\text{sel}}}{2.v \rightarrow \text{funcion}}$	$\frac{\frac{\text{obj}}{sumo(2.v)} \quad \frac{\text{funcion} \rightarrow \text{funcion}}{\text{sel}}}{sumo(2.v)(2.v) \rightarrow \text{funcion}}$
$\frac{b.f}{b.f \rightarrow \text{funcion}}$	$\frac{sumo(2.v)(2.v) \rightarrow \text{funcion} \quad b.f \rightarrow \text{funcion}}{\text{APP}} \quad \text{sumo}(2.v)(2.v) \rightarrow \text{funcion} \quad \text{sumo}(2.v)(2.v) \rightarrow \text{funcion}$		
$b.f(2.v) \rightarrow \text{funcion}$			

(\*)  $\frac{\frac{\text{obj}}{2 \rightarrow 2} \quad \frac{\text{funcion} \rightarrow \text{funcion}}{\text{sel}}}{2.v \rightarrow \text{funcion}} \quad ??$